HDM: A COMPOSABLE FRAMEWORK FOR BIG DATAPROCESSING

Ajitha Baby Fathima,
Assistant Professor CS Dept.,
St.John's College of Arts and Science,
xpfathima@gmail.com,
Kanyakumari.

Abstract: Big data is a large dataset that be analyzed computationally. Most of the IT inventions is going towards managing and maintaining big data. It also denoted for the large volume of data. The data may entered in both structured and unstructured format. It is mainly designed for gathering and storing large amount of data. Big data was mainly pointed with 3V, they are velocity, Volume and Variety. The complexity monitored in big data was optimizing the complicated jobs and pipeline optimization. In Existing the optimization of big data perform by manual thus it is time consuming and error prone. Thus in the proposed scheme basic data and task model. The novel model was implemented in the proposed scheme is HDM. It is a provenance based Application model. Used for History management and also proposed data parallel applications. The model also proposed the aggregation and filters for the high level monitoring sequence. The proposed scheme is maintained in the simple and efficient performing task. The main aim of the proposed model is used to improve the efficiency of the storage in cloud computing. Functional data dependency graph is maintained by the following features.

Keywords: Optimization, Big data, HDM, Aggregation, scheduling

INTRODUCTION

Big data is a term for data sets. It can handle large or complete data where the traditional data processing is inadequate to deal with them. The challenges include, capture, storage, analysis, data duration, search, sharing, transfer, visualization, querying, updating and information privacy. A database needs to be in order for the data inside of it be considered "big". Big data is the need for new techniques and tools in order to process those data. Getting programed on multiple machines to work together in an efficient way means big data is the one of the solution. Another advantage of big data is fast accessing. The distribute of data across a cluster and how those machines are network together are also important considerations which must be made when thinking about big data problems.

The uses of big data are almost as varied as they are large. Prominent examples you're probably already familiar with including social media network analyzing their members' data to learn more about them and connect them with content and advertising relevant to their interests, or search engines looking at the relationship between queries and results to give better answers to users' questions.

But the potential uses go much further! Two of the largest sources of data in large quantities are transactional data, including everything from stock prices to bank data to individual merchants' purchase histories; and sensor data, much of it coming from what is commonly referred to as the Internet of Things (IoT). This sensor data might be anything from measurements taken from robots on the manufacturing line of an automaker, to location data on a cell phone network, to instantaneous electrical usage in homes and businesses, to passenger boarding information taken on a transit system.

By analyzing this data, organizations are able to learn trends about the data they are measuring, as well as the people generating this data. The hope for this big data analysis are to provide more customized service and increased efficiencies in whatever industry the data is collected from.

One of the best-known methods for turning raw data into useful information is by what is known as Map Reduce. Map Reduce is a method for taking a large data set and performing computations on it across multiple computers, in parallel. It serves as a model for how to program, and is often used to refer to the actual implementation of this model.

In essence, Map Reduce consists of two parts. The Map function does sorting and filtering, taking data and placing it inside of categories so that it can be analyzed. The Reduce function provides a summary of this data by combining it all together. While largely credited to research which took place at Google, Map Reduce is now a generic term and refers to a general model used by many technologies.

Big data is a blanket term for the non-traditional strategies and technologies needed to gather, organize, process, and gather insights from large datasets. While the problem of working with data that exceeds the computing power or storage of a single computer is not new, the pervasiveness, scale, and value of this type of computing has greatly expanded in recent years.

In this article, we will talk about big data on a fundamental level and define common concepts you might come across while researching the subject. We will also take a high-level look at some of the processes and technologies currently being used in this space. The analysing tool for big data is known as Apache Hadoop. Apache Hadoop is a framework for storing and processing data in a large scale, and it is completely open source. Hadoop can run on commodity hardware, making it easy to use with an existing data center, or even to conduct analysis in the cloud. Hadoop is broken into four main parts:

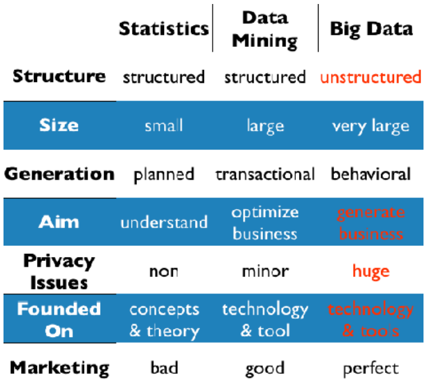
1. The Hadoop Distributed File System (HDFS), which is a distributed file system designed for very high aggregate bandwidth;
2. YARN, a platform for managing Hadoop's resources and scheduling programs which will run on the Hadoop infrastructure;
3. Map Reduce, as described above, a model for doing big data processing;
4. And a common set of libraries for other modules to use.

Other tools are out there too. One which has been receiving a lot of attention recently is Apache Spark. The main selling point of Spark is that it stores much of the data for processing in memory, as opposed to on disk, which for certain kinds of analysis can be much faster. Depending on the operation, analysts may see results a hundred times faster or more. Spark can use the Hadoop Distributed File System, but it is also capable of working with other data stores, like Apache Cassandra or Open Stack Swift. It's also fairly easy to run Spark on a single local machine, making testing and development easier.

EXISTING SYSTEM

The existing system was based upon the simple storing and performance of the area through which it could be maintained. The model framework is quite difficult for processing the data. The main aim of the proposed model is simple and efficient for the system can be maintained in the simple and storing performance through which it should be maintained. The main aim of the processing ability can be performed in the simple structure of performance of which it should be manipulated.

The manipulating images is further maintained by the simple accessing processing. The big data analysis is the part through which it should be used for the data mining concept. The major drawback that is analyzed in the system was processed in the sequence through which it should be fetched on the particular screening process. This process is based upon the fact through which the data are fetched into simple process. Here the processing abilities can be maintained under the simple concepts. The previous work is based upon the big data with simple application through which it should be manipulated. The data that stored in the server is not efficient to fetch the information. This information is simple and make it use of the previous usage through which it should be determined. The most obvious concept that behind the big data scheme is not vulnerable for the entire server. Data is the collection of variables and values. The size of the data is always increasing. Storing this data without using it will occupy the storage space waste. Data mining is analyzing data with different perception and summarizes it into useful information. Mining the information make the helps organizations make proactive, knowledge driven decisions and answer questions that were time consuming to resolve.



	Statistics	Data Mining	Big Data
Structure	structured	structured	unstructured
Size	small	large	very large
Generation	planned	transactional	behavioral
Aim	understand	optimize business	generate business
Privacy Issues	non	minor	huge
Founded On	concepts & theory	technology & tool	technology & tools
Marketing	bad	good	perfect

Fig 1: Statistical Report

Data Mining is also called as Knowledge Recovery in Data Base or knowledge discovery and mining is the process of automatically searching large volumes of data for patterns such as association rules. Data mining is important as the particular user will be looking for pattern and not for complete data in the database, it is better to read wanted data than unwanted data. Data mining task can be classified into summarization, classification, clustering, association and trends analysis. Summarization is the abstraction of generalization of data. Fig 1 statistical report shows the difference of using data mining and big data.

Classification is the method of classifying objects into certain groups based on their attributes. Association is the discovery of connection of objects. The association is based on certain rules known as association rules. These rules reveal the association of objects; they used to find the correlation of set objects. Clustering is the identification of clusters or groups whose classes or unknown. Clustering should be done such that the similarities between objects of different types are minimized.

Trend analysis is the matching of objects changing trends such as increasing streaks. Data mining tools can be classified into three categories they are tradition data mining tools, dashboards and text mining tools. Big data there is a lot of data that is growing on the web every day. The data has been so large that it becomes difficult to analyze it with the help of our traditional mining tools. Big data term is established before the crossing the ability it has three main characteristics volume, velocity variety.

Big data is a massive amount of digital data is collected from various sources that are too large and raw in form. Big data deals with the new challenges such as complexity, security, risks to privacy.

There are so many big data challenges to be considered they used to capture, storage, analysis, data curation, search, sharing, transfer, visualization, querying, updating and information privacy. This should be processe4d in the mechanism that should be solved in the system that can be maintained in the system to be used. This can be formularized this can be maintain in the system to which it should be used. The main aim of the big data analysis is used to increase the fetch efficiency mechanism. These mechanism is similar to the range through which it can be maintained in the system of which it should be used. The main aim of the big data analysis is based upon the similar concept of changes.

PROPOSED SYSTEM

Programming abstraction is the core of the networking model. Though which it should be processed by the model to which it should be determined. The processing ability of which it can be processed in the storage of the system it can be used for the simple and accessing of the model of which to be processed. The main use of the processing ability could be maintained a simple accessing tool of the DDDM (Dynamic distributed data matrix). The triple DM is the process of which it should be processed in the opinion of the summarization model of which it should be maintained. Here the process ability could be particularly overcome by the system of perfect methodology of allocation. This should be maintained by the performance of which it could be maintained in the system.

The DDDM process can be help to access the performance that is processed in the severe application of which it should be used for the certain applicable methodology. This can be further improvised and implemented in the sequence of the abstraction of model through which it can be determined. The model ability that focused on the simple and efficient performance of which it should be used for the several use of application to be determined.

Basically a DDDM is represented as the DDDM [I, O] in which I and O are data types for input and output. The DDDM itself transfer the function that change into Input to Output. Apart from the core attribute DDDM also contains information like data orientation, location, distribution to support optimization and execution. The structure ‘inType’ and outType’ is used to guarantee the type correctness during optimization and composition ‘category’ is for the differential variable. ‘Children’ and ‘dependency’ are used to reconstruct the DDDM during job planning and optimization. The attribute function is the ‘core function’ of DDDM. The data allocation model for the distributed function is designed below.

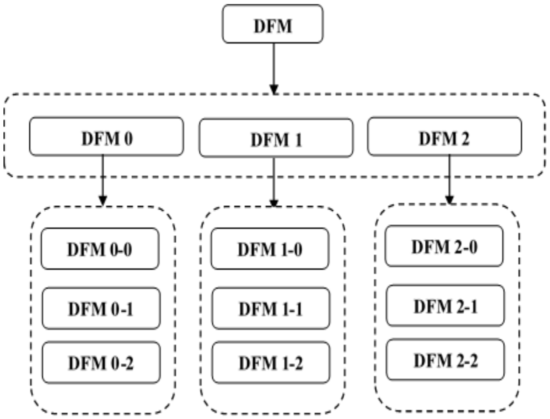


Fig: 2 Data Model of DDDM

The above design denote the simple and efficient performance of the data that should be particularly denoted for the best performance of allocation data. This concept is depend upon the simple and efficient time consuming of the performance of which it should be used for the best result. The main aim of the proposed model is used to overcome the time delay of which it can be mainly used for the performance of which it could be maintained in the system of which it should be used. The another model of the frequency to be used for the simple and accessing scheme of which it could be maintained in the system of the performance to which it should be denoted.

The Fig 2 contains the basic model of the system that could be performed in the data storage system of which it should be used. The DFM (Dynamic Functional Matrix) is mainly used for the performance of which it should be used for the best performance evaluation strategy. These are used for the functional dependency of the system to which it should be maintained in the simple and efficient strategy of storing data. It could be further modularized and applicable in the system of which it should be used for the best performance evaluation. The next each DFM contains is then derived to be DDDM for storing the dynamic efficient model of which it should be used for the performance of which it could be used for the best performance strategy. Then the group of each particulars could be rarely found to be the best of performance through which it could be determined of another strategic resource.

The main use of the systematic design of the performance of which it should be processed in the sequence of which it could be used for the systematic design. Another main use of the systematic design of the proposed model is depend upon the simple wise edition of the performance that should be design based upon the needs. This should be further improvised by the entire edition of the performance that is used to store the information based upon the simple and efficient file accessing. The big data simple and efficient time performance through which it should be used for the simple and efficient performance could be used for the fetching information in the simple and efficient of which it should be used.

The data that's performed in the simple performance for fetching the information in a different way of performance through different criteria. This can be performed in the simple and efficient way of knowledge that could be used for the performance of the design to which it could be designed for the fast data accessing

The proposed model is based upon the Map reduce scheme. This scheme is used for the performance of the job assigning strategy to which it should be used. This performance is not a simple task through which it can be maintained in the performance of the area to which we determined. This process of maintaining a system with efficient need of performance of the ability of the processing needs could be maintained in the different needs. These needs are performed in a way through which it could be designed for the job scheduling purpose. The job scheduling process is similarly consists of large amount of data in which it could be used for the further performance of which it should be used for the simple accessing ability. The performance of which it should be maintained in a similar field is not equality within process. These process are maintained in the area of which it should be used for further ability.

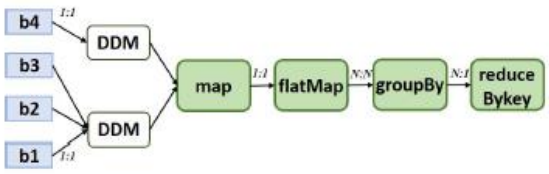


Fig 3 First data Flow

The first data flow of the job maintains in the model of the processor such as {b1, b2, b3, b4}. This can be further modified by the means of which it should be used for the several other performance through which it should be used for the proposed model of DDDM.

The DDDM accept the request and align the statement of performance of which it should be declared by the means of performance of which it could be maintained in the system. The DDDM is then transmit the data to the flatMap. The flatMap comes under the Mapping sequence of which the data should be stored in the similar sequence of performance through which it should be used for the several number of performance through which it should be used. The main use of the groupBy is based upon the performance of which it should be used for the several other process it could be used. Thus the map reduce is used to overcome the unwanted data that can be placed in the performance of which it can be maintained in entire system of performance through which it should be used. Another usage of the performance of the proposed model is based upon the clear performance of the system to be used. The main use of the proposed model can be used for the other systematic procedure through which it should be used for the entire scheme. Thus it can be maintained in the entire system of process of all the maintaining performance in the area through which it should be used.

Another use of the similar performance can be maintained by the other usage of the performance of which it should be used for the simple accessing model. The accessing model is used for the performance of which it should be used for the similar usage.

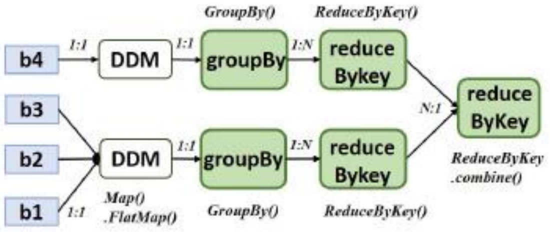


Fig 4: Local Aggregation Process

The main use of the proposed model is not simple process of which it should be used for the simple accessing process. This can be maintain by the scheme of which it should be used for the similar accessing capability. The main use of the local aggregation could be used for the accessing ability. The main use of the grouping could be maintained in the system should be maintained in the system of which it should be used for the simple concept.

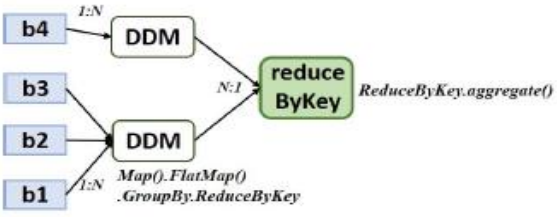


Fig 5: Fusion Model

The main use of the performance it should be maintained in the systematic procedures of which it could be used. The final stage is the fusion technology through which it could be used for the performance of which it should be used for the performance of which it could be used. The main usage of the system of little can be maintained in the system could be used for the performance of which it could be processed in the mechanism all the data could be used. The fusion model is used to combine the information in the simple sequence the data can be maintained in the similar concept of the performance to which it should be added and performed.

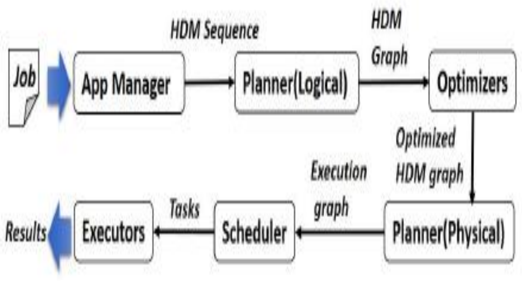


Fig 6: Process of executing DDDM jobs

The processing model was depend upon the sequence of performance of which it should be used for the similar stage of performance. There are two algorithms that should be used in the proposed scheme they are

1. Logical Plan Algorithm
2. Physical plan Algorithm

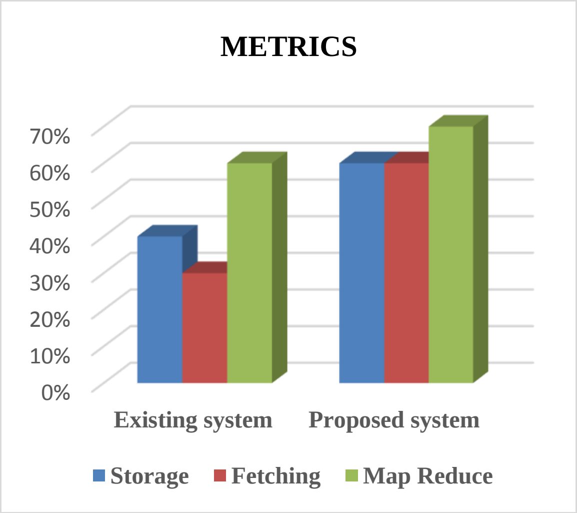
The logical plan algorithm is used to find the free space in the scheduling task. This could be either performed by the performance of which it should be used by the simple and efficient performance through which it should be used for the further usage of the system it could be performed. The performance engine could be used for the perfect and efficient roll back sequence of which it should be used

EXPERIMENETAL EVALUATION

The previous work is based upon the performance of which it should be used for the big data analysis. This can be maintain din the system of which it should be used for the simple performance of which it should be maintained in the system could be used. This is further processed in the simple concept of the data to be used for the performance of data fetching. The query fetching and storage of data is not familiar with the previous scheme of which it should be used for the performance of the data that should be used on big data an analysis to overcome the error. The main use of the proposed model is depends upon the simple usage of the systematic use of the query fetching option in the proposed scheme. This is the scheme of which it should be used for the particulars of which it should be used for the smart storage and getting the data.

The algorithm that proposed in the previous scheme was Logical and Physical plan algorithm of which it should be used for the systematic behavior of the scheme to be proposed.

Thus the proposed algorithm and design successfully fetch the information without any time delay and also the data that should be fetched and stored in the proposed model is simple and provide 75% user friendly application of which it should be used.



The proposed model is efficiently processed in the system of which it could be comparably processed in the proposed work of performance of which it should be used for the efficient data processing. The proposed model is high efficiently support the information of all the performance of which it could be used. The main use of the performance of the system could be maintained in the storage, fetching and map reduce concept is concerned and maintain in the Map Reduce concept.

CONCLUSION

The proposed model implement the performance of the job scheduling and map reduce concept with efficient manner of performance of which it should be used. Thus the above model could be maintained in the simple algorithm process to which it could be used for the similar concept. This concept will effectively performed in the DDDM concept. The DDDM and DFM is used for the efficient data accessing process. This can be maintain under the performance of which it should be used for the simple accessing scheme it could be maintain

Further in future this project is implemented in the disk based processing needs for the performance, consider the fault tolerance model, the main use of the project should be used for the similar concept. One long term problem to be used for the optimization work.

REFERENECE

1. Alexander Alexandrov, Rico Bergmann, Stephan Ewen, Johann- Christoph Freytag, Fabian Hueske, Arvid Heise, Odej Kao, Marcus Leich, Ulf Leser, Volker Markl, Felix Naumann, Mathias Peters, Astrid Rheinl¨ander, Matthias J. Sax, Sebastian Schelter, Mareike H¨oger, Kostas Tzoumas, and Daniel Warneke. The Stratosphere platform for big data analytics. VLDB J., 23(6), 2014.
2. Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, and Matei Zaharia. Spark SQL: Relational Data Processing in Spark. In SIGMOD, pages 1383–1394, 2015.
3. Craig Chambers, Ashish Raniwala, Frances Perry, Stephen Adams, Robert R. Henry, Robert Bradshaw, and Nathan Weizenbaum. FlumeJava: easy, efficient data-parallel pipelines. In PLDI, 2010.
4. Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. Commun. ACM, 51(1), 2008.
5. Yin Huai, Ashutosh Chauhan, Alan Gates, G¨ unther Hagleitner, Eric N. Hanson, Owen O'Malley, Jitendra Pandey, Yuan Yuan, Rubao Lee, and Xiaodong Zhang. Major technical advancements in Apache Hive. In SIGMOD, pages 1235–1246, 2014.
6. Mohammad Islam, Angelo K. Huang, Mohamed Battisha, Michelle Chiang, Santhosh Srinivasan, Craig Peters, Andreas Neumann, and Alejandro Abdelnur. Oozie: towards a scalable workflow management system for hadoop. In SIGMOD Workshops, 2012.
7. Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In SIGMOD Conference, 2010.
8. Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In SIGMOD, 2008.
9. Bikas Saha, Hitesh Shah, Siddharth Seth, Gopal Vijayaraghavan, Arun C. Murthy, and Carlo Curino. Apache Tez: A Unifying Framework for Modeling and Building Data Processing Applications. In SIGMOD, 2015.
10. Sherif Sakr and Mohamed Medhat Gaber, editors. Large Scale and Big Data - Processing and Management. Auerbach Publications, 2014.
11. Sherif Sakr, Anna Liu, and Ayman G. Fayoumi. The family of mapreduce and large-scale data processing systems. ACM CSUR, 46(1):11, 2013.
12. D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, and Michael Young. Machine learning: The high interest credit card of technical debt. In SE4ML: Software Engineering for Machine Learning, 2014.
13. ChunWei Tsai, Chin Feng Lai, Han Chieh Chao, and Athanasios V. Vasilakos. Big data analytics: a survey. Journal of Big Data, 2(21), 2015.
14. Dongyao Wu, Sherif Sakr, Liming Zhu, and Qinghua Lu. Composable and Efficient Functional Big Data Processing Framework. In IEEE Big Data, 2015.
15. Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. In NSDI, 2012.

Authors:

Miss. Ajitha Baby Fathima received her MCA from Anna University, and M.Phil degree from Noorul Islam University. Presently she is a Ass. prof in Department of Computer Science. Her Research interests include cloud computing, Networking, Image Processing, .